Aleksandra Kolobaeva

QUALITY CONTROL IN GPCR PROTEIN FAMILY LABELING AS A DATA MINING PROBLEM

Knowledge engineering for bioinformatics

Bachelor thesis

Information Technology

2017

Kaakkois-Suomen ammattikorkeakoulu

Author (authors)	**Degree**	**Time**
Aleksandra Kolobaeva	Bachelor of Engineering	March 2017
Thesis Title Quality control in GPCR protein family labeling as a Data Mining problem Knowledge engineering for bioinformatics		41 pages 0 pages of appendices
Commissioned by SOCO Research Group		
Supervisor Reijo Vuohelainen		

Abstract

Biology and medicine are becoming strongly data-dependent sciences where advances are, more than ever, based on data acquired by sophisticated machinery and methods. One area in which this is especially true is bioinformatics. Bioinformatics deals with –omics data, including proteomics, which was the field of the current project.

This study addressed quality control of curated protein databases, and can, therefore, be considered as a knowledge engineering problem.

G protein-coupled receptors are a large super-family of cell membrane proteins of interest to biology in general and pharmacology in particular. One of its families, class C, is of specific interest to pharmacology and drug design. This family is known to be quite heterogeneous and the discrimination of its several sub-families is a difficult problem, as it must rely on their primary amino acid sequences. We were interested not as much in investigating sub-family discrimination using a standard classification approach *per se*, but in exploring sequence misclassification behavior. To be more precise, we used well-known data mining classification techniques to isolate sequences that were very often misclassified and almost always, that is, consistently, to the same wrong sub-family.

I hope that this work will be a useful step towards assisting protein database curators in their quality control duties by providing them with knowledge of management tools.

Keywords

Machine learning, G protein-coupled receptors, knowledge engineering, bioinformatics

TABLE OF CONTENTS

1 INTRODUCTION

Nowadays, data is becoming a pervasive presence in most spheres of human life and, day after day, their availability grows exponentially. Nevertheless, the availability of data by itself does not entail the availability of information, while the latter does not entail we have generated knowledge. It could even be argued that the generation of knowledge could not be considered to be a goal as such, unless such knowledge becomes actionable in practical terms. That is, until it can be used in practical terms.

Acquiring knowledge of data is by no means a trivial task and requires careful analyses to find trends, patterns and anomalies in datasets in order to make decisions based upon them. Powerful computing approaches and a sound mathematical base are required for performing all these tasks, which help people to make informed deductions from raw numbers. This could be said the to be core task of data mining (Hall, Witten, & Frank, 2011). Data Mining (DM) can be loosely defined as the science of extracting useful knowledge of often large and not always homogeneously structured data repositories, involving methods at the intersection of artificial intelligence, machine learning, statistics, and database management systems.

These days, few areas of knowledge are experiencing such a huge shift towards data-based knowledge extraction or DM as biology and medicine. They are both very quickly becoming strongly data-dependent sciences, where advances are, more than ever, based on data acquired by sophisticated machinery and methods (Halka, 2014).

One of the main drivers behind this trend are the extremely fast changes created by the advances in genomics. They have, in fact, coalescing with the also very quick advances in computer sciences (CS), all but created a whole new field of science in which data-dependency is especially true, namely Bioinformatics. Bioinformatics deals, using CS methods, with what have come to be known as –omics data, including genomics, proteomics, metabolomics and transcriptomics.

Proteomics is, precisely, the specific field of interest of the current project, which addresses the quality control of curated protein databases as a specific instance of DM and can, therefore, be considered as a knowledge engineering problem.

G protein-coupled receptors (known as GPCRs) are a large super-family of cell membrane proteins. They have of late become a hot topic in bioinformatics due to their interest to biology in general and pharmacology in particular. One of its several families, namely class C, is of specific interest to pharmacology and drug design. This family is known to be quite heterogeneous and the discrimination of its several sub-families is known to be a far from trivial problem, as it can only be addressed through the analysis of their primary amino acid sequences (Bräuner-Osborne, Wellendorph, & Jensen, 2007).

One straightforward type of DM analysis in this scenario would entail the classification of these receptors into their many families and sub-families using supervised classifier techniques. From the onset, I was interested not as much in investigating sub-family discrimination using a standard classification approach *per se*, but, instead, in exploring sequence misclassification behavior. To be more precise, I used well-known data mining classification techniques to isolate sequences that were very often misclassified and almost always, that is, consistently, to the same wrong sub-family.

The reason for deciding to use this approach is that, in this way, I could begin to address a key problem in protein database management, which is quality control in receptor labelling. We must bear in mind the family/subfamily assignment of these proteins is still very much work-in-progress and that there is not complete agreement about it within the scientific community. There is thus room for data-based analyses that could shed light into this difficult problem.
I hope that this work will ultimately be a useful first step towards assisting protein database curators in their quality control duties by providing them with data-based knowledge management tools.

This project is circumscribed to current and ongoing research developed at the Soft Computing (SOCO) Research Group, part of the Computer Science department at *Universitat Politècnica de Catalunya* BarcelonaTech (UPC). This research is formally funded by the Spanish *Ministerio de Economía y Competitividad* (MINECO) under project grant TIN2012-31377. The project named "*KAPPA AIM: Knowledge Acquisition*

in Pharmacoproteomics usingmAdvanced Artificial Intelligence Methods", led by Dr. Alfredo Vellido, specifically investigates class C GPCRs using statistical and Machine Learning (ML) techniques.

As part of KAPPA AIM, previous work by PhD student Caroline Konig and Drs. Alfredo Vellido and René Alquézar from SOCO, investigated the problem of Class C GPCR sub-family discrimination as a supervised problem using Support Vector Machine (SVM) classifiers. It was concluded that, beyond an agreeable level of classification accuracy, some of the class labels assigned to GPCR sequences of this class by curated databases were, at the very least, questionable, due to the consistency of their misclassification behavior across many instantiations of the classification process; a consistency that can be expressed as a tendency for a given sequence with a database-assigned label to be predicted mostly as belonging to a differently labeled subtype. Experiments were restricted to SVM classifiers, within a cross-validation procedure, for a given type of data transformation. Through these experiments, a shortlist of "strongly misclassified cases" was generated (König, Cárdenas, Giraldo, Alquézar, & Vellido, 2015).

The objective of the project is a first approximation to the target of moving beyond SVMs to discover whether alternative classifiers provide equally consistent misclassification results. This way, we could start assessing to what extent the misclassification behavior is classifier-dependent or not.

The structure of the document is as follows: after the current introduction, Chapter 2 provides the reader with the necessary basics of the biological background for the DM analysis. It includes a description of the GPCR proteins and their relevance to biological processes and pharmacology tasks, as well as a summary description of the particular data under analysis in the project. This is followed, in Chapter 3, by a self-contained introduction to the DM techniques and methods employed for the analyses. Such analyses and their results are then reported and discussed in Chapter 4. Chapter 5 wraps up the report by providing summary conclusions and some directions for possible future research.

2 BIOLOGICAL BACKGROUND

2.1 Bioinformatics and data science

Biology and medicine are becoming strongly data-dependent sciences, where advances are more than ever based on data acquired by sophisticated machinery and methods. One area in which this is especially true is bioinformatics. This sub-field of the biological sciences is very inter-disciplinary and can be more or less vaguely defined by the development of methods and computer-based software and analytical tools for understanding biological data in general. It combines computer science, statistics, mathematics, pattern recognition and engineering to analyze and interpret biological data in general. Bioinformatics mostly deals with –omics data, including proteomics, which is the field of the current project (Perco, et al., 2006).

Analyzing proteomics data to extract usable knowledge from them is what this study is about. The analyses involve using pattern recognition and DM techniques and methods.

It has only recently been fully acknowledged that a series of breakthroughs in medical science and information technology are triggering a convergence between the healthcare industry and the life sciences industry that will quickly lead to a fully redesigned set of relationships among patients, their doctors and biopharmaceutical companies, mediated by information technology and innovation in the –omics sciences (Burns, 2012).

The current study focuses on a very specific problem in bioinformatics and relates to the area of proteomics for pharmacology. In particular, data from a type of cell membrane proteins called G Protein-Coupled Receptors (GPCRs) were analyzed, which are the target of a large number of recently developed drugs (George, O'Dowd, & Lee, 2002). This chapter is focused on a brief and not-too-biologically detailed description of GPCRs and of the specific set of data analyzed is in the following chapters.

2.2 GPCR superfamily

G Protein-Coupled Receptors are highly important proteins in life of plants, fungi and, most importantly for human beings, mammals as homo sapiens species belong to this class. There are approximately 1,000 different G Protein-Coupled Receptors in the body of human and each of them dedicated on a special aim. Every year scientists find new separate purposed receptors and these findings open new opportunities for pharmacology and therapy. As a result, health and well-being of human beings are affected by drugs with higher accuracy.

GPCRs come into action with such functions as odor and taste sensing, vision, and operation of central nervous system, which are, undoubtedly, necessary for life-sustaining activity of every stand-alone individual. Receptors are located in the cell membrane and play crucial role in cell functioning and communication with the external environment and other cells. The main task of them is to sense specific compounds (agonist), which activate signal generation to the inner part of the cell thereby triggering a reaction inside to regulate cell functioning. In general, agonists have two types: endogenous and exogenous. Endogenous agonists are compounds originated from the organism or tissue (neurotransmitters, hormones), while exogenous are of external nature (in this case drugs). When endogenous do not operate in a proper way, exogenous are entrusted to substitute endogenous agonists and regulate activity of the cell. Agonist connected to GPCR cause chemical reactions, what is called signal relay cascade. Signal relay cascades represent chain reaction of chemical charges which differs in complexity although target the same communication event: change state of the cell through sending signal from one to another. Common goal of all G protein-coupled receptors. In such a manner, it may affect cell on various levels as behavior and structure of cell or activity of particular enzymes inside.

There are huge diversity of agonists and, thus, all the superfamily of GPCRs are separated into several classes based on purpose and, consequently, location. Overall, GPCR superfamily contains 6 classes named from A to F, while at the same time these classes include subclasses and this hierarchy goes even deeper, dividing

proteins into groups from the perspective of functioning and structure. Classes are ordered as follows:

1. Class A – Rhodopsin-like receptors.
2. Class B – Secretin receptor family.
3. Class C – Metabotropic glutamate receptors.
4. Class D – Fungal mating pheromone receptors.
5. Class E – Cyclic AMP receptors.
6. Class F – Frizzled and Smoothened receptors.

Even though GPCR vary dramatically in types of chemicals that activate signal process, common tendency still recognizable within each class. So, rhodopsin-like GPCRs are mainly represent photon detecting receptors, responsible for all kind of light sensing functions in rod cells (night vision). Secretin receptors work with secretin hormone, the common goal of which is to regulate water balance mainly in the digestive organs. Representatives of class C sense neurotransmitters (chemicals allowing neuron communication) and operate basically in brain. Task of class D is clear from the name: communication between mates and sexual reproduction of fungi. Cyclic adenosine monophosphate (cAMP) is secondary agent for conduction signals inside cells via class E receptors for hormones, that are not able to enter through cell membrane. Class F of GPCRs is the less studied yet and contains two kinds of agonist: frizzled and smoothened proteins working with various signaling pathways, which firstly were discovered in flies' genome (Drosophila, for instance), however important for human beings as well. This description of diversity of G protein-coupled receptors types provides with a glimpse of importance and need for research in the field of proteomics. Proper classification allows making clearer conclusions about every type of receptor and, as a result, creating much more well-targeted medications. In this work, we concentrate on class C – metabotropic glutamate receptors. In this work, we concentrate on class C – metabotropic glutamate receptors.

Class C of GPCRs play huge role in the functioning of CNS (central nervous system). Generally, class C is targeted to the neurotransmitter called glutamate – chemical used for signal conduction between neural cells, still few exceptions exist. There are 7

subclasses of class C GPCR and in majority databases they are numbered in the following order:

1. Metabotropic Glutamate
2. Calcium Sensing
3. GABAB (gamma-aminobutyric acid)
4. Vomeronasal
5. Pheromone
6. Odorant
7. Taste

Synaptic plasticity and depression: new insights from stress and rapid-acting antidepressants

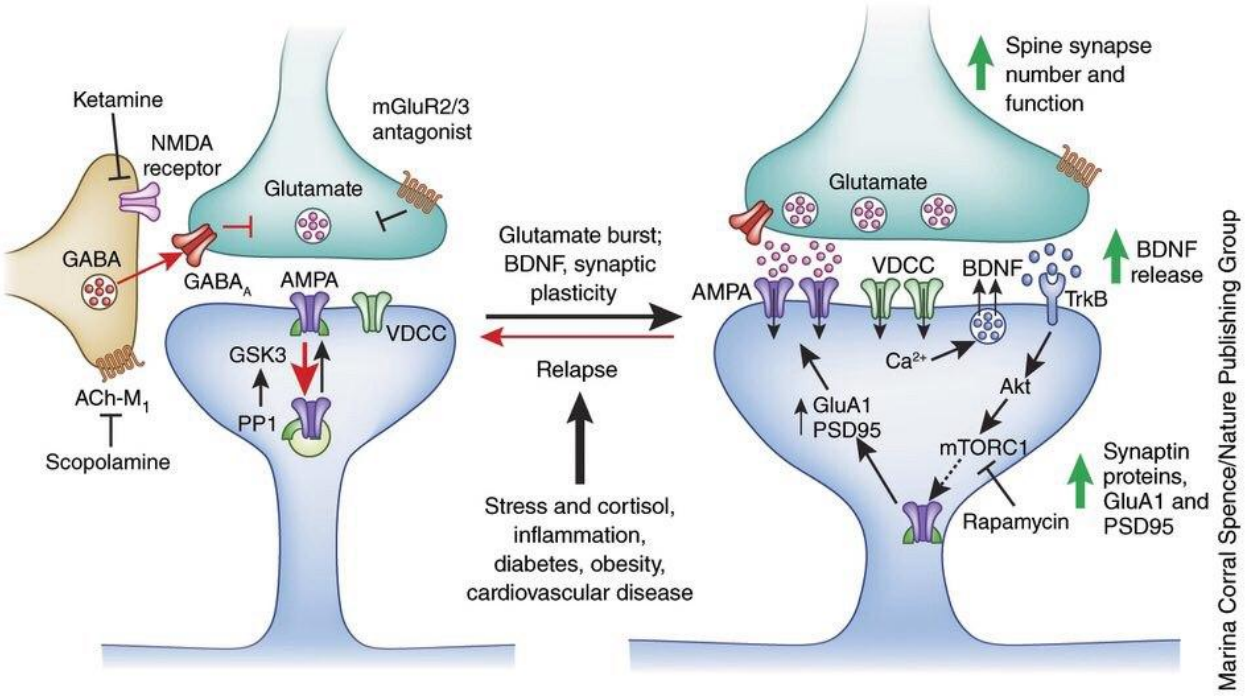

Figure 1 Overview of GPCR acting between synapse (Ronald S Duman, 2016)

Metabotropic Glutamate and GABAB located in synapses. Synapse is a conjunction between neural cells, which is responsible for signal conduction, more specifically flowing neurotransmitters with help of chemical reaction from one cell to another cell's receptors. These subclasses represent special interest for human health as they interact with drugs during treatment of neural system diseases such as Parkinson,

Alzheimer, drug addiction, schizophrenia, anxiety, etc. Calcium Sensing reacts to the change of calcium ion concentration outside the cell. Calcium Sensing receptors located in glands and kidneys of mammals and play enormous role not only in skeleton diseases treatment like osteoporosis, but even asthma and some cases of cancer (Pollak M.R., 1994). Vomeronasal, Pheromone, Odorant and Taste receptors have a similar function and located in sensory neurons in epithelium of nasal cavity, tongue, and palate. In far as is concerned animals, these proteins act a crucial part in search of non-toxic nutrients, choice of correspond mates and danger identification. In food production, variety compounds are used to affect taste receptors in order to deliver the desired effect like increasing flavor or relieving bitterness of the products (Mombaerts, 2004).

From the previous paragraph it is well-marked that subclasses of class C GPCR are of different nature and operate with own group of agonists. The structure varies as well from class to class. Common GPCRs structure is represented as seven spiral-like segments called helices, that pierce cell membrane and connected with each other in series (called 7 trans-membrane helices) by the loops (3 outside and 3 inside the cell). In addition, two significant parts of the protein are termini: one in the outer part of the cell called N-terminus and one inside called C-terminus. N-terminus senses agonist and forwards signal to the inner part, while main function of C-terminus is to conduct signal to the cell and interact with G protein in cytoplasm, causing required reaction on agonist. The highest difference from class to class is noticed in TM5, TM6 and N-terminus.

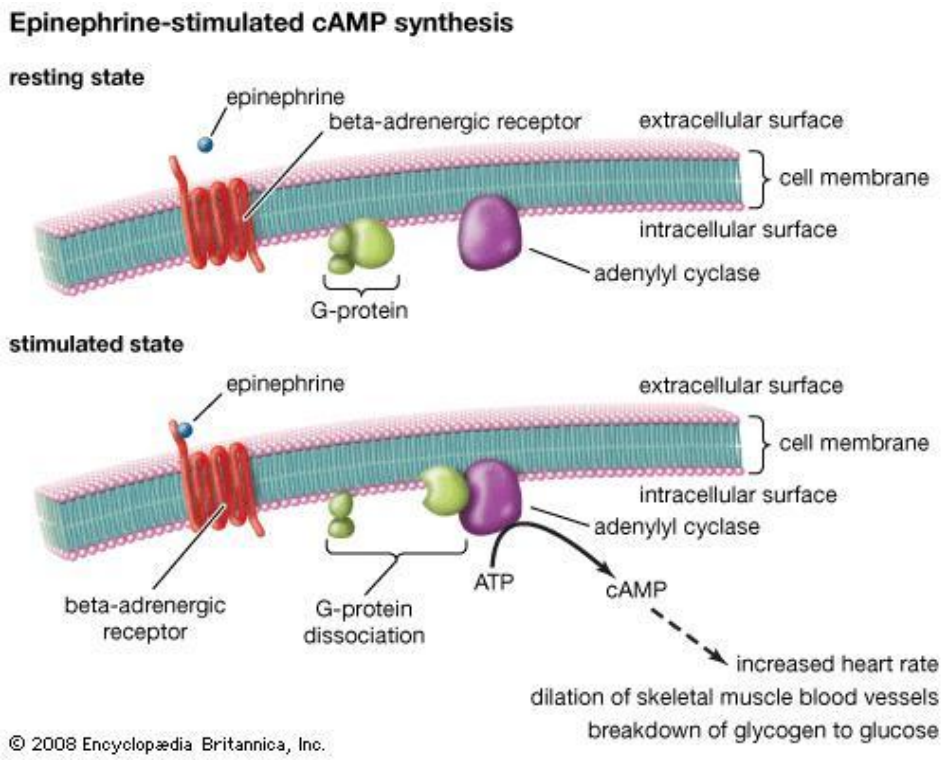

Figure 2 Overview of GPCR structure (Krantz)

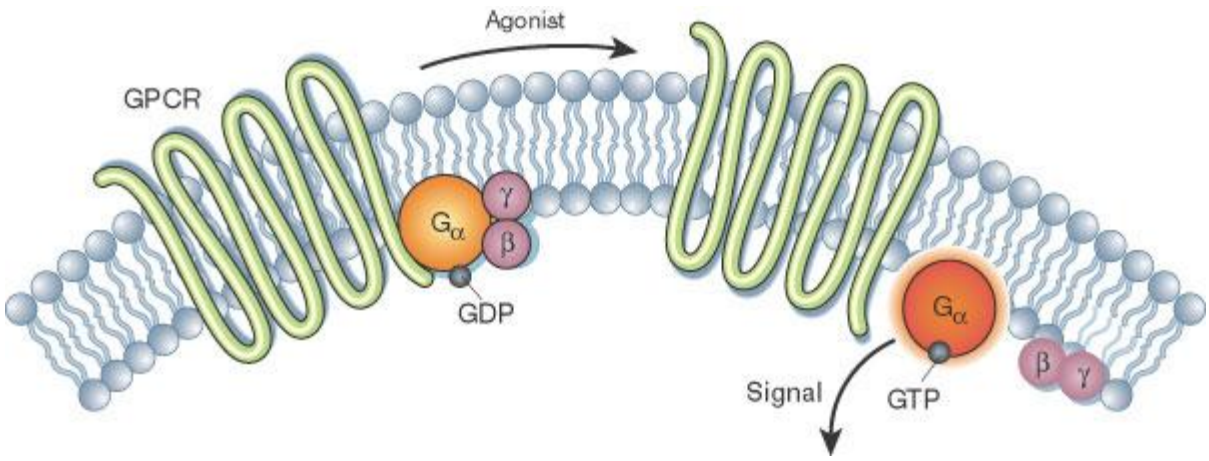

Figure 3 Agonist binding and G protein reaction (Li, 2002)

The picture is taken from the dophamine of class A research and it represents a case

from class A GPCR with mutations. As all GPCRs are relatively similar in a structure, this image is a good example for explanation of amino acid sequences in class C GPCRs.

All GPCRs are similar in the structure: each of them consists of 7 trans-membrane segments called helices and two tails called «terminus». There are two types of helices: one outer N-terminus, which senses ligands, and one inner connected to G protein.

2.4 Description of the analyzed data

There is almost no knowledge about the full crystal 3-D structure of GPCRs. Only very recently, some partial GPCR structures have been solved, and most of them are not from the Class C in which we are interested in this study, but from class A.

For class C, unfortunately, no full crystal 3-D structure has yet been solved; only two TM domains and several extra-cellular domains have been described in. This means that, in order to investigate these receptors, we must rely on the analysis of their primary structure, expressed as an amino acid symbolic sequence. The good news here is that class C primary sequences are publicly available from several curated international databases.

mGlu receptors are activated by glutamate, a major excitatory neurotransmitter in the brain,a nd they are involved in neurological disorders including Alzheimer's and Parkinson's diseases, Fragile X syndrome, depression, schizophrenia, anxiety, and pain. The CaS receptor is activated by the calcium ion and it is known to play a key role in extra-cellular calcium homeostasis regulation. GB is a neurotransmitter that mediates most inhibitory actions in the central nervous system; it is involved in chronic pain, anxiety, depression and addiction pathologies.

A total of 1,510 class C GPCR sequences (from GPCRDB version 11.3.4, as of March 2011) belonging to these seven sub-families are analyzed in the experiments reported in the following chapters. Their actual distribution of cases by sub-family is summarized in Table 2.

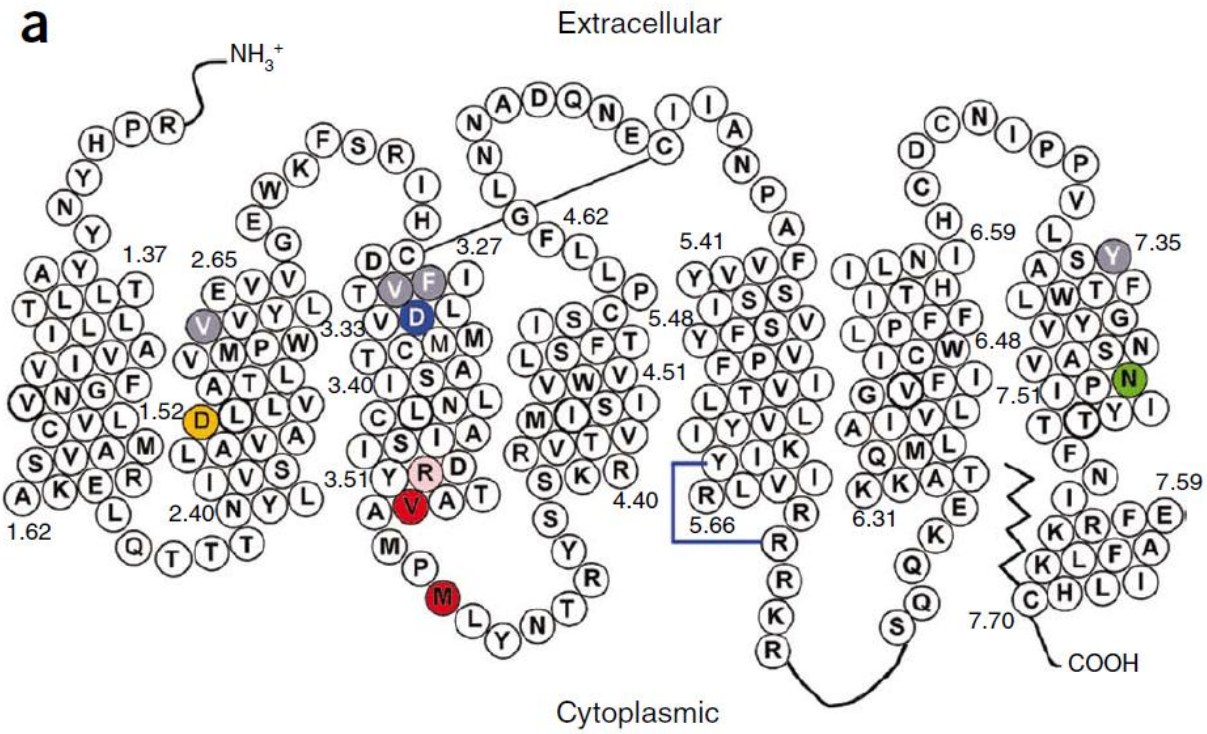

Figure 4 Amino acid sequence in GPCR structure (Yang Han, 2009)

Table 2: Number of GPCRDB class C available sequences for each of the seven GPCR Class C sub-families (Shkurin & Vellido, 2016).

Sub-family	♯ sequences
Metabotropic Glutamate	351
Calcium Sensing	48
$GABA_B$	208
Vomeronasal	344
Pheromone	392
Odorant	102
Taste	65

The primary sequences cannot be analyzed as symbolic amino acid arrays using most standard statistical or pattern recognition methods, and, therefore, they have to be transformed for investigation. Many transformations have been suggested in the literature and several of them were considered for our experiments.

The first one uses directly the 20 amino acids (see Table 3) of which the GPCR sequence alphabet consists. Despite its simplicity, its use has previously resulted in surprisingly solid performances.

Table 3: Table of the 20 amino acids that are present in the GPCR symbolic *alphabet* (Shkurin & Vellido, 2016).

AA name	symbol	AA name	symbol	AA name	symbol
Alanine	A	Glycine	G	Proline	P
Arginine	R	Histidine	H	Serine	S
Asparagine	N	Isoleucine	I	Threonine	T
Aspartate	D	Leucine	L	Tryptophan	W
Cysteine	C	Lysine	K	Tyrosine	Y
Glutamate	E	Methionine	M	Valine	V
Glutamine	Q	Phenylalanine	F		

Subsets of amino acids may share similar physico-chemical properties, which makes them equivalent at a functional level. Amino acid grouping also helps computations by reducing the dimensionality of the analyzed data set. For this study, two alternative groupings were used, in the form of sub-sequence frequencies (see Table 4): the Sezerman (SEZ) alphabet (11 groups) and the Davies Random (DAV) alphabet (9 groups).

Amino acids and their groupings were not just used as such in this study, but in the form of n-grams, which are subsequences of length n. The concept of n-grams is well-known in protein analysis. Here, we used the relative frequencies of the n-grams. Therefore, the n-gram representation consists on the relative frequency of each n-gram in a sequence (note that for Sezerman and Davies, the length of the n-gram is not taken in number of amino acids, but in number of groupings). Due to the exponential growth of the size of n-grams, experiments were limited to n-grams of size 1, 2 and 3.

3 TECHNICAL BACKGROUND

3.1 Classifiers

Classification could be generically defined as the task of learning a target function *f* (classification model) that maps each attribute set *X* to one of the predefined class labels *y*. The input data for such classification task is a collection of items (also alternatively defined as records, instances or examples, mostly depending on the particular application area we are dealing with) and characterized as *X* and *y*. Therefore, *X* represents features of an object and *y* – target or class label assigned to such object (where such assignment can be the result of human decision or of an automated or semi-automated process). Unlike in regression and prediction tasks, classes must be defined as discrete in classification tasks.

A classifier is therefore nothing but a method, technique or algorithm for building a class-discrimination or classification model based on a sample of input data. A classifier deploys a learning algorithm (using the machine learning terminology) to create a model which fits best dependencies between inputs and targets (class labels). Moreover, the model should ideally be capable of fitting new data that was not observed before (that is, in the creation of the classification model itself). In other words, a classifier model is *trained* by fitting *training data* to *training targets* so that it can optimally predict the class labels of unseen *test* data (Steinbach, Kumar, & Tan, 2004).

It must be acknowledged from the onset that, at this time, the palette of different classifier techniques available to the data analyst are almost overwhelming and that there are not soundly established guidelines about the suitability of specific choices. To be more precise, the suitability of certain classifiers seems to be mostly problem dependent, so that very simple classifiers suffice for certain problems, whereas much more sophisticated ones are required for others.

Classifiers of any kind must be evaluated, so that the definition of pertinent performance evaluation measures is also of utmost importance, as they can explain the quality of the built classification models. Given that the main goal of this work is the analysis of misclassifications, I decided to use two main performance measures: classification accuracy and confusion matrices. Accuracy is a very standard measure

that reflects the ratio or percentage of correctly classified cases. It is very useful for comparing different classifiers, but it is also important to understand that it only reflects overall performance. Accuracy, in percentage form, is calculated like this:

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions} \times 100$$

It is important to note that, if accuracy is 50% or less, it is fair to say that the evaluated classifier is useless given that its predictions are less than chance (Asker & Maclin, 1997). The second performance evaluator, the confusion matrix, displays the correct and incorrect entries of *N* classes in an $N \times N$ matrix. The confusion matrix is thus far more detailed than the accuracy and provides us with a full picture of the classifier's correct classification and misclassification behaviour. Examples of confusion matrices are available in section 4.2 (Experimental results).

3.1.1 Decision Trees

A Decision Tree (DT) is a dendrogram-like, branching classifier consisting of a hierarchically arranged set of nodes that describes a problem with multiple possible solutions, showing the relationship of different events as well as the probability for each event sequence. Typical DTs include the following types of nodes:

- Decision (Root) – no incoming and 0 or more outgoing edges; represented by squares
- Chance (Internal) – 1 incoming edge and 2 or more outgoing edges; represented by circles
- End (Leaf or Terminal) – one incoming edge and no outgoing; represented by triangles

In a DT, each of the terminal nodes contains one class value; the rest of nodes does not contain any of class as they represent condition to distinguish objects which have different features (See an example of the graphical representation of a DT in Figure 5).

The estimation of the class label of a test observation starts at the root of a DT. Going through the tree, the test observation must relate to one of the decisions of each test condition and follow the appropriate tree edge accordingly. Once it reaches the leaf

(the “bottom end” of the tree), the corresponding class label is assigned to the observation.

A DT model can be made more interpretable by linearizing it in the form of decision rules, where each rule describes the descriptions leading to each of the leafs of the tree. This takes the form of the following statement: *If* **rule_1** *and* **rule_2** *and* **rule_3** *and* … *and* **rule_N** *then* **outcome A**.

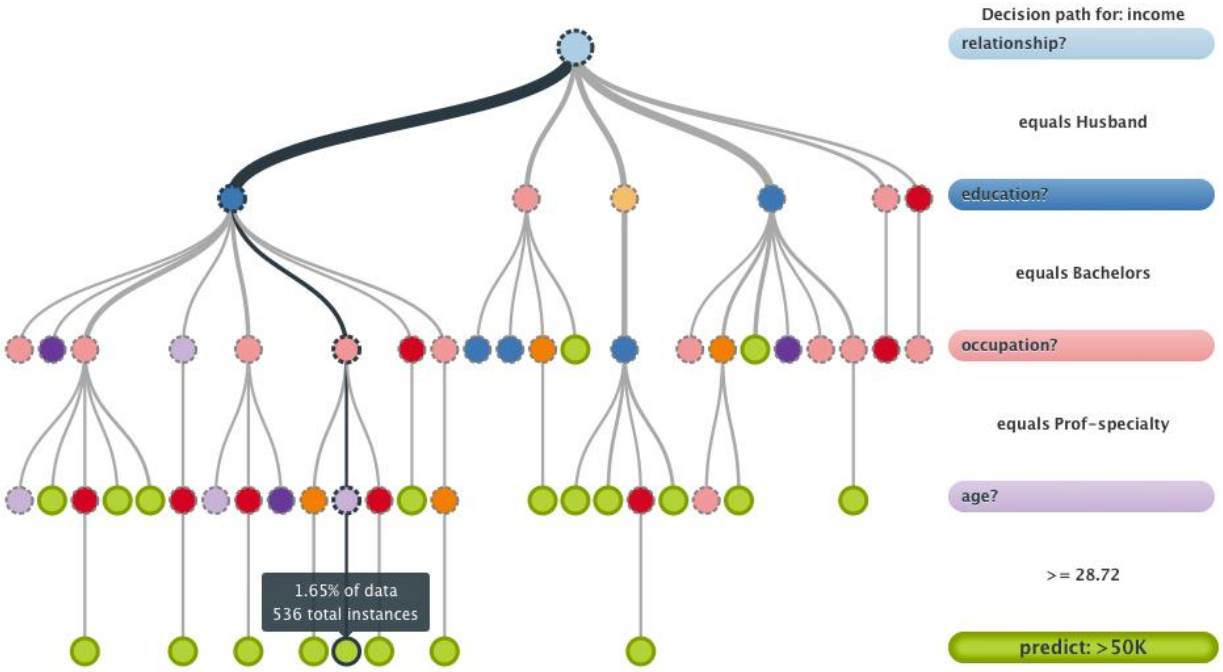

Figure 5: Example of Decision Tree representation.

Sometimes, the features of the dataset may be of different types. There maybe three main types of them in a DT: categorical, nominal and continuous.

Paying attention to attribute type is important because each of them have own characteristics is treated in different ways. The first one, categorical variables, are usually described as variable having several categories, but with no established ordering. An example is the eye color attribute, containing several colors as variables (e.g. blue, green, brown) and no intuitive ordering. Test condition may include as many splits as amount of answer variants. However, some algorithms like CART (Classification and Regression Tree) use only binary splits to build a tree by

considering $2^{k-1} - 1$ ways for *k* attribute values in the test condition. This is realized by grouping values into two sets; size does not matter.

The second type, nominal variables comprise several values with an obvious sequence ordering. A trivial example would be the size of clothes (Small, Medium, Large, etc). Binary and multiway splits are utilized in this case with a rule of proper ordering.

Finally, continuous attributes might be organized in binary style with comparison test ($A \leq v$) or ($A > v$) with further comparison. Multiway condition obeys the rule $v_i \leq A < v_{i+1}$, for $i = 1, \dots, k$ so that all the answers are considered (Asker & Maclin, 1997).

DTs use different metrics to choose the best answer at each condition node to know the quality of a particular branching (split). Three of these metrics are called *Gini Impurity*, *Entropy* and *Classification Error*.

If after splitting of parent node class distribution of child node is even (50% for one of two classes), then impurity is maximum. The lowest impurity (zero impurity) appears when node shows exactly one class belongs to the answer or 100% probability. Here $p(k|t)$ or just p_k be set of objects belonging to a class k at node t. Metrics equations as follows:

$$Gini = 1 - \sum_{k=1}^{K} p_k^2$$

$$Entropy = \sum_{k=1}^{K} p_k^2 \log p_k$$

$$Classification\ error = 1 - \max_k p_k$$

To estimate the performance in a test setting, gain is measured. Gain is the difference between parent node before splitting and child node after splitting. Mathematically, gain is expressed as:

$$\Delta = I(parent) - \sum_{j=1}^{k} \frac{N(v_j)}{N} I(v_j),$$

where I is the given node's impurity, *N* is the amount of observations of parent node, *k* is amount of attributes and $N(v_j)$ is amount of abreacts related to given node.

3.1.2 *K*-Nearest Neighbors

k-NN is one of the simplest classifiers in data mining. However, it shows high results in various applications. Often *k*-NN is associated with *lazy learning* concept, main idea of which lies in classification without minimal or no preliminary conclusions about training data, what signifies fast training. Instead, all the training samples are used during test phase. Training cases allocated on a n-dimensional vector of features called feature space. To assign a test case to some class, *k* nearest training cases are estimated. Nearness is calculated using *metric* – function to calculate distance between cases.

The standard *k*-NN formulation uses Euclidean distance as the metric to define similarity between observations and their neighbors. It must be noted, though, that there exist many other distance metrics apart from the Euclidean one (e.g. Manhattan, Minkowski); their choice often depends on the nature and type of data. Here is a formula for Euclidean distance, where *d* is a distance, *x* and *y* are the cases:

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

Each time a new model is built, *k* needs to be recalculated as it affects accuracy: if *k* is too small, noise will impair performance, whereas if it is too big, boundaries between classes may become blurry. For example, in a task with two classes *k* is recommended to be an odd value, so that an ambiguous classification situation cannot arise (Kellller, Gray, & Givens, 1985).

3.1.3 Quadratic Discriminant Analysis Algorithm

Quadratic discriminant analysis (QDA) is a statistical algorithm based upon idea of class separation by quadratic combination of variables. This classifier is a variation of linear discriminant analysis (LDA). The idea of such “upgraded” algorithm arose because of the limitations of LDA in multi-class problems as it requires data to be normally distributed on feature space in order to minimize error of class prediction.

QDA allows building more complex class separating surfaces in comparison to LDA. This fact means that the boundaries between different classes are quadratic in nature. This leads to the possibility of obtaining more accurate results in multi-class problem. Unlike linear distances, quadratic distances are not symmetric. If the determinant of the group covariance matrix is less than one, the generalized squared distance can be negative. QDA allows for the heterogeneity of classes' covariance matrices, what means higher amount of data types may be used for classification using this method.

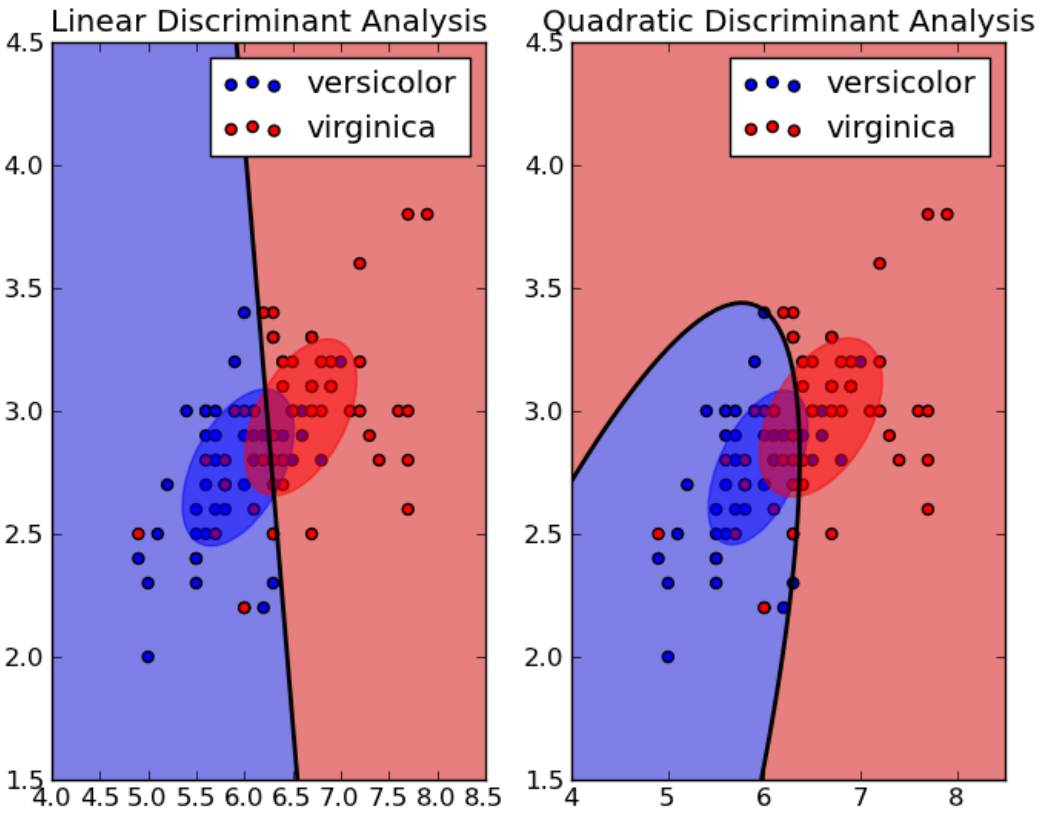

Figure 6: Comparison of LDA and QDA with typical test data “Fisher’s Iris data set” (Scikit, 2010)

3.2 Software tools

All the classifiers used in this study (a very small sample of the possible choices) has a long story of practical application to real research problems. All the reported experiments in the next chapter were completed using algorithms implemented in the Matlab language and using its version R2015b. Matlab (short for Matrix Laboratory) is

a computer environment and fourth-generation programming language. This program allows performing computations and analyzing of multidimensional data. It is proprietary software and requires a license, which was gained from “Introduction To MATLAB Programming” course from coursera.org (Tairas, Fitzpatrick, & Ledeczi, 2015). The design of Matlab aimed to allow programmer to write powerful code in a compact manner and with a strong focus on matrix operations.

The choice of programing software is due to analyst skills and experience, as well as in its adequacy to the available data format and its visualization capabilities. The *Classification Learner* application is perfect for the comparing results of several models.

3.3 Cross-validation

Cross-validation (CV) is a method to guarantee the fairness of the predictive performance of a mathematical model (in our case, a classification model). While there are several different types of CV (e.g. leave-one-out and k-fold), the main idea remains the same: the whole data is separated into reciprocally exclusive sets: a usually larger one (which is the training set with which the model is actually created) and a usually smaller one (the validation set, not used to create the model and therefore acting as a test). The training data is used to create a model, while the test data is used to validate the model, i.e. the ready model is applied to the test set and compare the results to the real values (as appear in the test set). This process is then iterated with all subsets, until each observation in the data set is used at least once for the test set (Kohavi, 1995).

In this work k-fold (10-fold to be precise) CV was used for all the classifiers. For this CV technique, one fold is chosen randomly for testing the model and the model is formed with rest of the *k*-1 folds. This process is repeated *k* times, so that each fold works once as validation data.

To create the final estimation of a new model, results of each fold are combined into an average value. The main advantage of *k*-fold CV over other alternative types is the participation of each fold in data validation at least one time, which means that as

much data as possible affects the resulting model and makes it accurate to the limit (Schneider, 1997).

3.4 Confusion matrix

In supervised machine learning very classification algorithm has a performance, which requires to be evaluated and summarized in order to make conclusions about quality of predictions. There is variety of ways to do it, but the most common is confusion matrix also called error matrix. Common accuracy of algorithm may be misinforming as it because common number may seem high, however practically be low in some classes thereby hiding the problem. Confusion matrix gives better understanding of what is done right and what kind of errors exists.

Confusion matrix (matrix for classifier with n classes denoted from 0 to n-1) appears like a table n by n sections, possibly containing such values as accuracy, error-rate, precision, etc. Let us have 2 classes and a classifier predicting belonging of each case to one of classes. Then, confusion matrix of classification looks as in Table 1:

Table 1Example of Confusion matrix

	$y = 1$	$y = 0$
$\hat{y} = 1$	True Positive (TP)	False Positive (FP)
$\hat{y} = 0$	False Negative (FN)	True Negative (TN)

Here ŷ is the answer of classifier for the case and y is a true class label for an object in a database. So, four results appear:

- TP: correct positive
- FP: incorrect positive
- TN: correct negative
- FN: incorrect negative

Consequently, there are two types of error False Negative and False Positive. From these results, some basic measurements about classifier are counted: error rate and accuracy. Error rate is a sum of false results FP and FN divided by all results (TP, FP, TN, FN). Accuracy, on the contrary, is amount of all true predictions TP and TN divided by common number of results. As well there are such measures like sensitivity, specificity, precision, False positive rate and various correlations, that are calculated from results of confusion matrix.

4 EXPERIMENTS

In this chapter, we first provide some summary details about the settings of the classifiers for the reported experiments. This is followed by a report of the experiments themselves: first reporting the results and then briefly discussing them and drawing some conclusions that, ultimately, set the path towards an approach that might be useful for proteomics experts and GPCR database curators.

4.1 Experimental settings

As was mentioned before, GPCR data is available in the three versions (that is, the original primary sequences of amino acids have been transformed in three different ways so that they can be straightforwardly used in our classifiers of choice): Amino acid (AA), Sezerman and Davies, with different encoding of the protein sequences; parts of code that appear often in the protein sequence are substituted with a letter and further combined to n-grams of size 1-to-3.

In previous research, data from GPCRDB underwent feature selection (GPCRDB partnership, 2007). Feature selection is the process of automatic selection of attributes in the data that are most relevant to the predictive modeling problem (in this case, Class C GPCR sub-family doiscrimination). In this previous research, the highest accuracy was achieved by a subset of Sezerman data with a result that was significant for 14 out of 21 binary classifiers. This is the reason we used these data for all three classification algorithms in the results reported next (Konig, Vellido, Alquezar, & Geraldo, 2014).

Each of the classifiers requires some level of individual adjustment and setting. Here, the DT classifier used Gini's diversity index split criterion because it best suits the current classification task. *K*-Nearest Neighbors was trained with $k = 5$ as the optimal number of neighbours. Distance metric used was the Euclidean, with *distance weight* set to *equal*. In Quadratic Discriminate Analyzer, a diagonal covariance regularization matrix was chosen.

4.2 Experimental results

I first report the tables with the average cross-validation accuracies for each of the three data transformations (AA in Table 4, Davies in Table 5 and Sezerman in Table 6). Best results are highlighted in bold.

Table 4: Average CV accuracy results for all three classifiers, for the AA Class C GPCR primary sequence transformation.

Classifier	*Accuracy*
Decision Tree	83%
k-Nearest Neighbors	**90.1%**
Quadratic Discriminant	82.6%

Table 5: Average CV accuracy results for all three classifiers, for the Davies Class C GPCR primary sequence transformation.

Classifier	*Accuracy*
Decision Tree	73.1%
k-Nearest Neighbors	**89.1%**
Quadratic Discriminant	85%

Table 6: Average CV accuracy results for all three classifiers, for the Sezerman Class C GPCR primary sequence transformation.

Classifier	*Accuracy*
Decision Tree	78.9%
k-Nearest Neighbors	**90.4%**
Quadratic Discriminant	85%

I now move to the report of the tables describing the confusion matrices for each of the three data transformations and for each of the three classifiers. Note that the rows of the matrices indicate the *true* class label of each GPCR sequence as it stands in the analyzed database, whereas the columns correspond to the *predicted* class labels according to the classification model (the classes are the Class C GPCR seven subfamilies described in Chapter 2, namely: 1: mGlu, 2: CaSR, 3: $GABA_B$, 4: VN, 5: Ph, 6: Od, 7: Ta).

The percentage values in the diagonal and in the right hand-side column are those of correct classification (e.g., the average classification accuracy for mGlu is 87.7%). The bottom row results of the right hand-side column, as can be easily guessed, are the percentages of wrongly classified sequences of each sub-family (True Positive Rates – TPR, and False Negative Rates – FNR). All percentages in each row add up to 100%. The different hues of pink in progression from white to red provide intuitive colour coding for these percentages.

The confusion matrices for the AA data transformation are reported in Figures 6, 7 and 8.

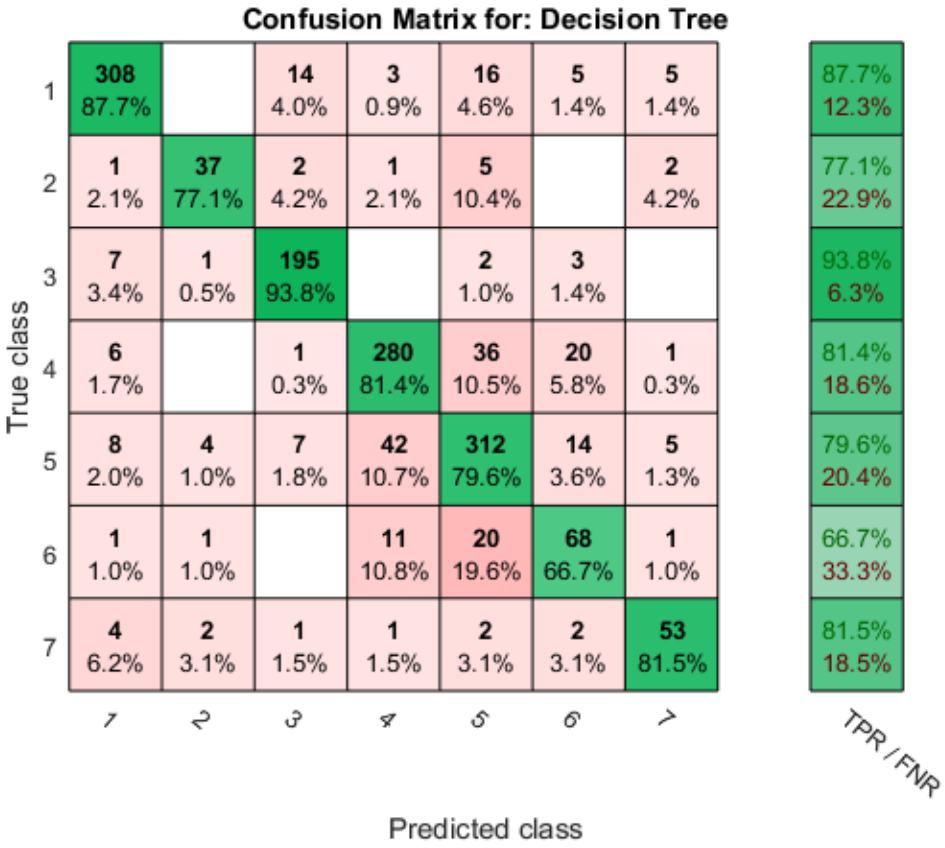

Figure 6: Confusion matrix for the AA data transformation, for the DT classifier.

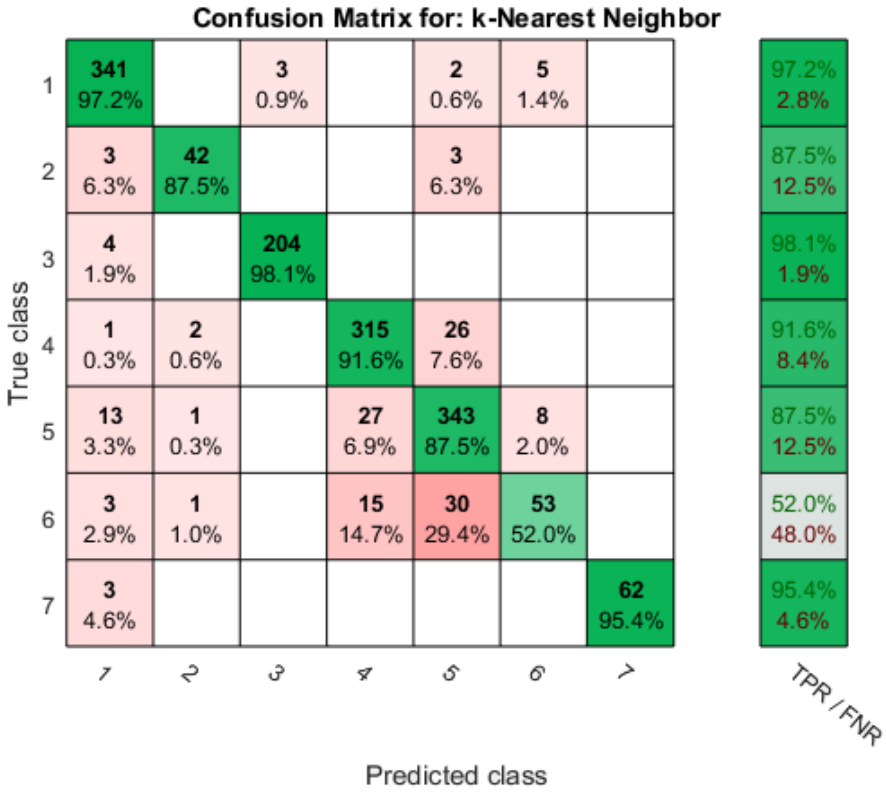

Figure 7: Confusion matrix for the AA data transformation, for the *K-NN* classifier.

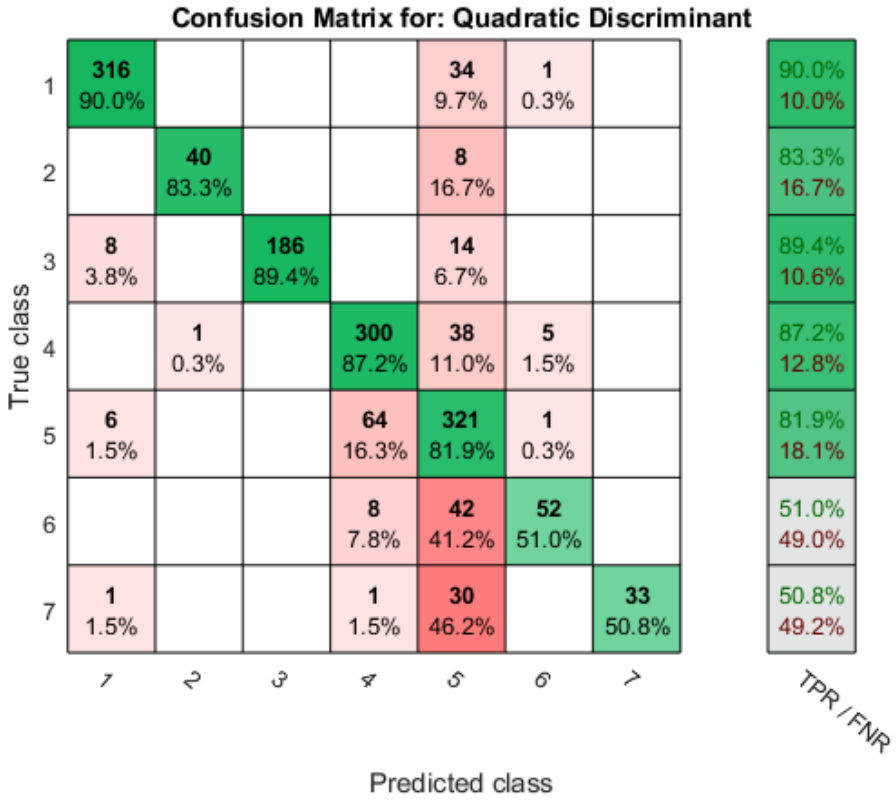

Figure 8: Confusion matrix for the AA data transformation, for the QDA classifier.

The confusion matrices for the Davies data transformation are reported in Figures 9, 10 and 11.

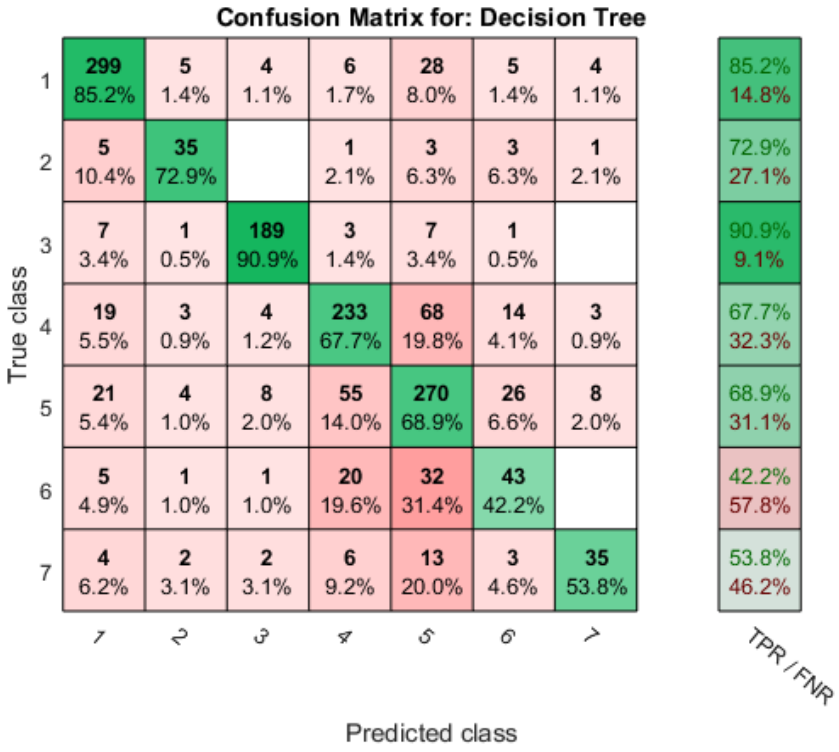

Figure 9: *Confusion matrix for the Davies data transformation, for the DT classifier.*

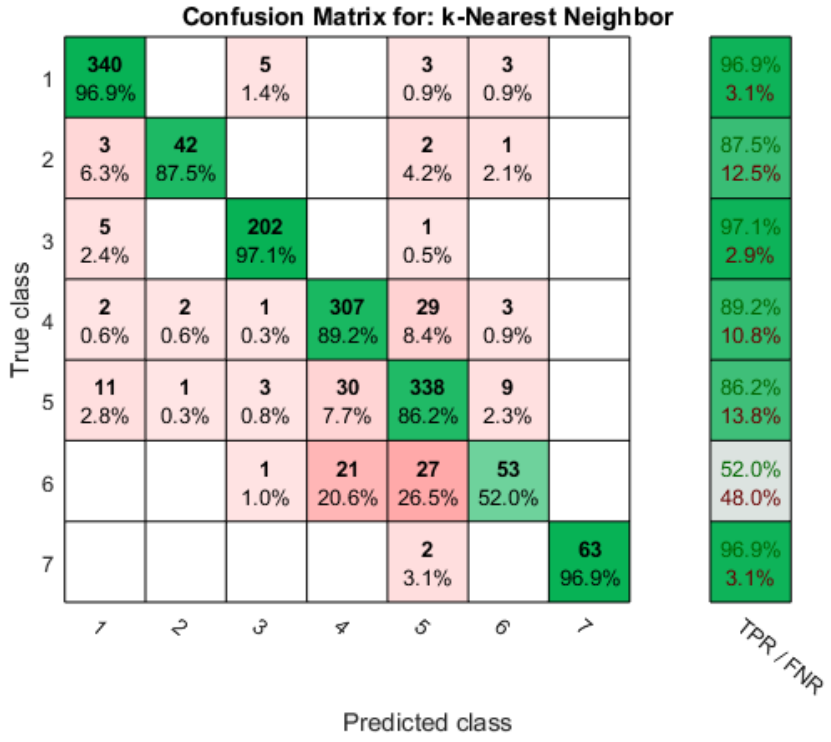

Figure 10: *Confusion matrix for the Davies data transformation, for the K-NN classifier.*

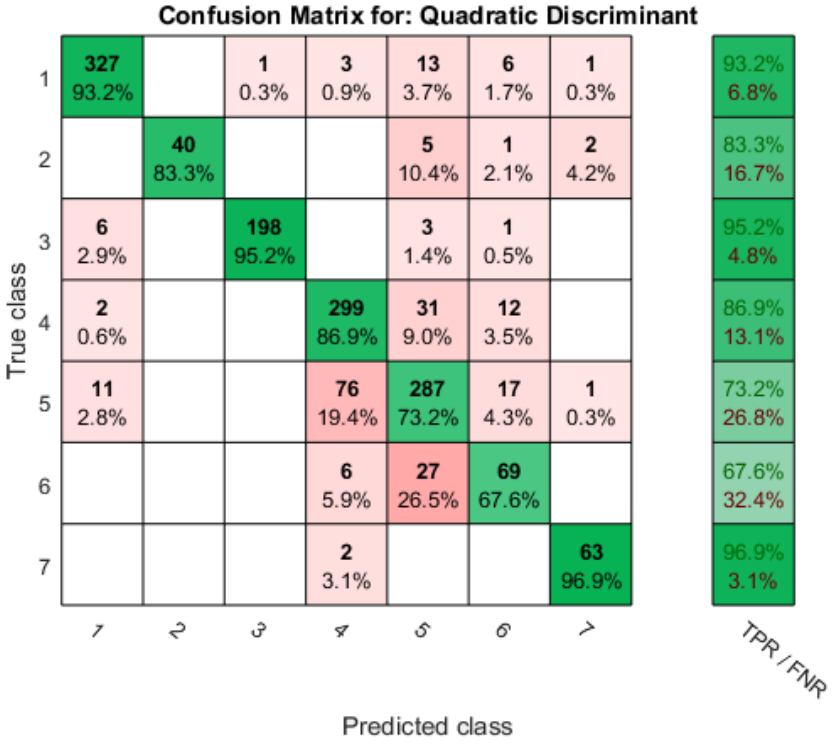

Figure 11: *Confusion matrix for the Davies data transformation, for the QDA classifier.*

The confusion matrices for the Sezerman transformation are reported in Figures 12, 13 and 14.

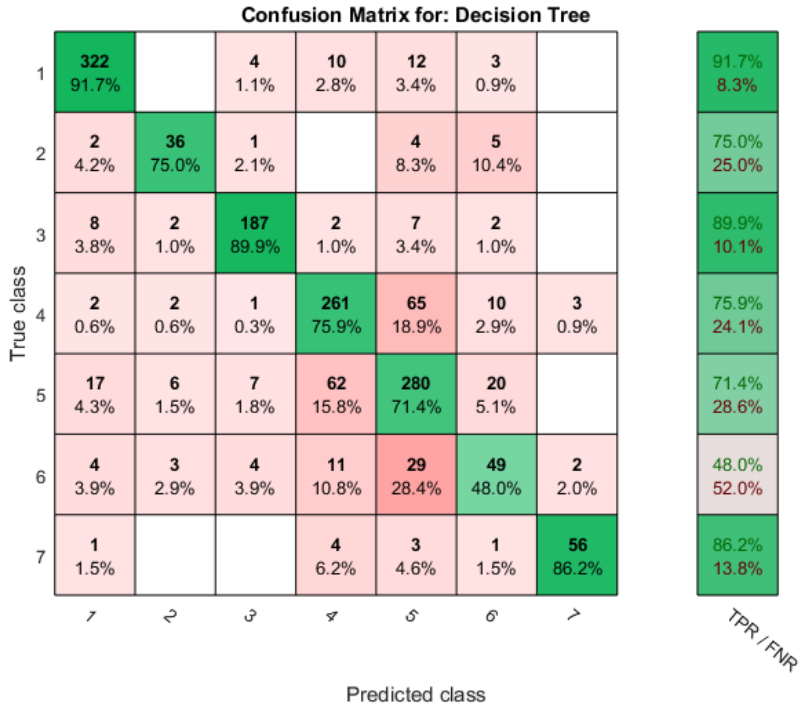

Figure 12: *Confusion matrix for the Sezerman data transformation, for the DT classifier.*

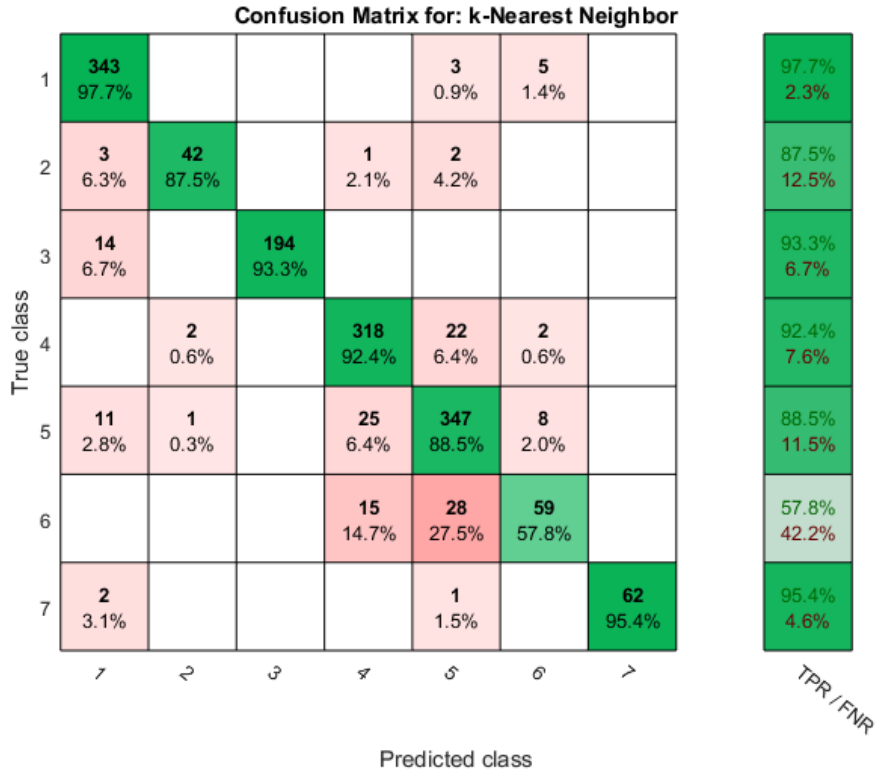

Figure 13: *Confusion matrix for the Sezerman data transformation, for the K-NN classifier.*

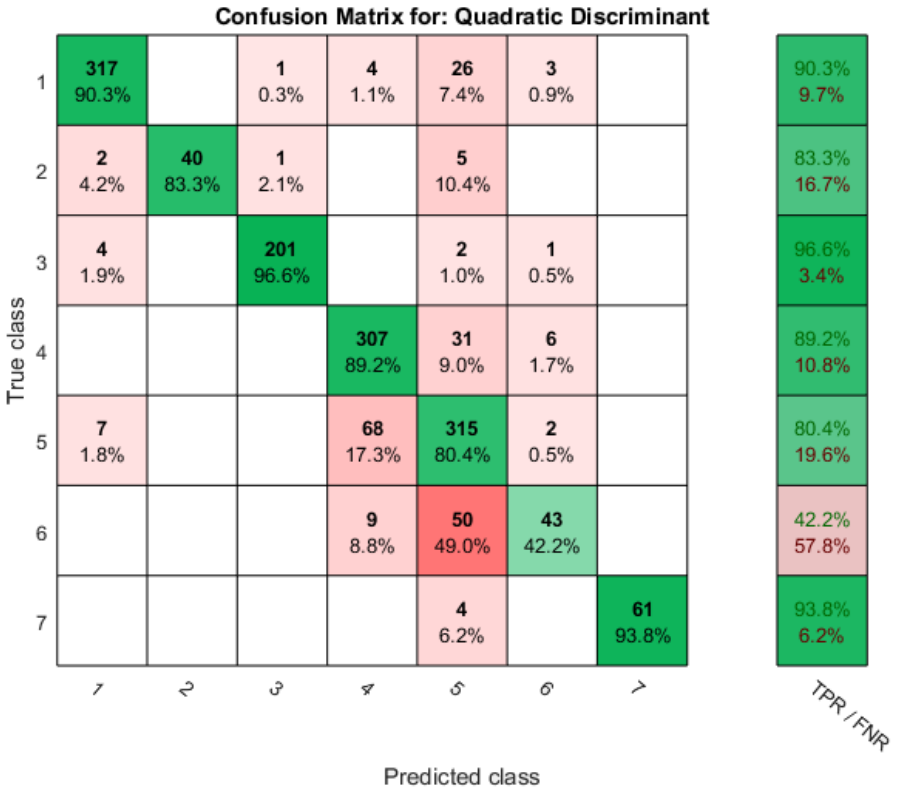

Figure 14: *Confusion matrix for the Sezerman data transformation, for the QDA classifier.*

4.3 Discussion of experiments

4.3.1 Overall cross-validation accuracy

The results reported in Tables 4, 5 and 6 are quite self-explanatory. For all data transformations, *K*-NN seems to yield substantially better results than either DTs or QDA, making it clearly the classifier of choice if only considering overall classification results. Note though that, given that we are dealing with a scenario of seven classes (Class C GPCR sub-families), this overall result might mask possible heterogeneity between the results of individual classes.

The differences between the results obtained from different data transformations for the same classifiers are somehow less conclusive. The Sezerman transformation seems to have a slight edge over the Davies transformation, and both seem better than the AA transformation. This is an important result for both Davies and Sezerman as it provides justification for their choice, especially given the fact that both transformations are extremely parsimonious.

4.3.2 Confusion matrices

We now move from the overall classification accuracy measure to the much more nuanced confusion matrices. They provide us with a more complex and, as a result, less straightforward interpretation of results.

We will start this time analyzing results by classifier.

For DTs, the confusion matrices corroborate the bad overall accuracy results. This is so because of the great spread of misclassifications across subfamilies. That is, for a given class (GPCR sub-family) some cases are misclassified as belonging to any of the other classes. This only happens with this classifier. The variations according to data transformation are notable but quite coherent: The best classified sub-family is always $GABA_B$ (93.8% accuracy for AA, lowering to 90.9 for Davies and 89.9 for Sezerman), whereas the worst classified is always Od (66.7% for AA, and as low as 42.2 for Davies and 48% for Sezerman)

Interestingly, and despite the spread of misclassifications, they do not seem to appear at random, in the sense that certain classes seem to be misclassified mostly as certain others: Throughout transformations, many Vn are misclassified as Ph and Ph as VN; many Od, in turn, are misclassified as either Vn or Ph. For some of the transformations, we must also add CsR and Taste misclassified as Ph. All in all, Ph seems a very heterogeneous sub-family generating considerable confusion in the classification process.

For *K*-NN, the picture is only partially different. Firstly, because the confusion matrices are much sparser. That is, sequences of each class are only misclassified as belonging to a much more restricted number of other classes. For the AA transformation, three sub-families reach the 90% accuracy threshold, namely Ta (95.4%), mGlu (97.2%) and $GABA_B$ (98.1%, the highest sub-family-specific accuracy achieved by all classifiers throughout transformations) and with the only bad behaving exception of Od (52%). Similar results are found when using the Davies transformation, with Ta and mGlu reaching a matching 96.6%, and $GABA_B$ a 97.1%. For Sezerman, Vn also reaches a substantial 92.4%, while $GABA_B$, Ta and mGlu reach, in turn, 93.3%, 95.4% and 97.7%.

For this classifier, the main causes of stable misclassification are quite concentrated in cases of Vn misclassified as Ph, cases of Ph misclassified as Vn and cases of Od misclassified as either Vn or Ph.

Finally, it is for QDA that we observe the clearest differences between the results according to the data transformation used. To be more precise, the results for AA are consistently poor, with only mGlu just reaching a 90% accuracy, while Od and, surprisingly, Ta, hardly reach the 50%. For this transformation, the Ph sub-family is clearly creating havoc, as many instances of other sub-families are estimated by QDA to actually be Ph cases. Davies and, especially, Sezerman transformations yield quite better results, mostly due to the far better classification of Ta. With Sezerman, three sub-families again reach the 90% accuracy barrier, namely mGlu (90.3%), Ta (93.8%) and $GABA_B$ (96.6%).

The consistent misclassifications are mostly the same as in previous classifiers but, in this case, an impressive 49% of Od cases are misclassified as Ph.

4.3.3 A shortlist of consistent misclassifications

The previously reported results provide us with a rather strong indication, if not evidence, that misclassification across classifiers and data transformations is by no means at random. That is, although some misclassifications maybe due to small variations on the two sides of slightly varying classificatiole n thresholds, many seem to hint that certain class labels are, at least, questionable.

Arguably, these “stubbornly” misclassified sequences would merit a closer look by database curators, in order to find out the causes of this apparent mislabeling behavior.

For this, in Table 7 we shortlist and individually identify by their database tags those Class C GPCR sequences that were most consistently misclassified as belonging to a different sub-family to that to which the case is supposed to belong according to its formal database label.

Table 7: Shortlist of the most consistently misclassified sequences across classifiers and data transformations. The alphanumeric code on the left hand-side column is their formal database identifier.

name	*DB class*	*predicted*
a8dz71_danre	1	6
a8dz72_danre	1	5
q5i5c3_9tele	1	5
XP_002740613	2	1
b0uyj3_danre	5	1
XP_002940566	6	5

My work could end up right here with the provision of a list of suspiciously labeled cases as the one presented in Table 7, and that would be the starting point of the work of a database curator or an expert in proteomics or bioinformatics. Nevertheless, bioinformatics is a field in which, fortunately, public database resources abound. For this reason, we can at least try to find what some internationally recognized protein databases (such as UniProt (www.uniprot.org), NCBI RefSeq

(www.ncbi.nlm.nih.gov/refseq), or Ensembl (www.ensembl.org), to name a few) say about these particular Class C GPCRs.

Most interestingly, some of these sequences have already been pinpointed as likely cases of mislabeling in other studies, exactly with the same type of misclassification. For instance, *q5i5c3_9tele* and *b0uyj3_danre* were identified in both (König, Cárdenas, Giraldo, Alquézar, & Vellido, 2015) (Shkurin & Vellido, 2016): the first has been described as a putative pheromone receptor[1], while the second appears as an uncharacterized and unreviewed protein[2]; *XP_002740613*, in turn, was identified in (König, Cárdenas, Giraldo, Alquézar, & Vellido, 2015) as an extreme error; while *a8dz71_danre* and *a8dz72_danre* were identified in (Shkurin & Vellido, 2016) and appear, again, as uncharacterized proteins[3] [4].

5 CONCLUSION AND FUTURE WORK

The GPCRs represent a major group of cell-surface receptors. These receptors play an important role in many physiological processes. These is a large and heterogeneous receptor family that has revealed itself as a major target in the design of therapeutic drugs. In fact, the growing knowledge of GPCRs and their ligands enables accelerating new drug design strategies (Klabunde & Hessler, 2002).

This study that now concludes has focused on a family of GPCRs, Class C, for which close no nothing is known about their three-dimensional structure, which is usually the basis of the investigation of their functionality. On the absence of such knowledge, we have used difference transformations of their amino acid primary sequences.

These primary sequences are available from public data repositories, out of which GPCRDB, the one I have used, is the most popular one specifically devoted to GPCRs, and is widely used by pharmacological companies and research biologists.

Very often, there is no “gold standard” for the assignment of a GPCR sequence to a specific sub-family of the many ones that exist, and such assignment is often based in

[1] http://www.uniprot.org/uniprot/Q5I5C3
[2] http://www.uniprot.org/uniprot/B0UYJ3
[3] http://www.uniprot.org/uniprot/A8DZ71
[4] http://www.uniprot.org/uniprot/A8DZ72

model predictions. In such scenario, database curators could benefit from the availability of tools that identify potential cases of dubious sub-family assignment. In this study, we have treated this problem as one of misclassification analysis and we have shown how we can isolate and identify consistent misclassifications.

This is, of course, only a very preliminary proposal that has helped me to developed my knowledge in different ways:

- It has allowed me to enter a field of research completely unknown to me beforehand which is bioinformatics, with its biology foundations.
- It has also allowed me to have a first glimpse of how research in this area is actually carried out.
- It has provided me with training in the one of the areas of artificial intelligence, namely Machine learning and has helped me to familiarize with some of its tools and techniques, as well as with specific programing languages and algorithmics.
- It has shown me the path for the use of analytical techniques as tools for knowledge engineering.

There are indeed many ways in which this preliminary research could be extended:

- Different classifier techniques could be used or could be added to the already investigated ones.
- Different and more complex data transformations could be used to compare its results with the ones reported here.
- Alternative GPCR databases could be explored.

All these tools could be standardized in the form of a software tool that could be made available to database curators and the bioinformatics community at large.

LIST OF REFERENCE

Asker, L., & Maclin, R. (1997). IJCAI-97: Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence . Nagoya: Professional Book Center.

Bräuner-Osborne, H., Wellendorph, P., & Jensen, A. A. (2007). Structure, pharmacology and therapeutic prospects of family C G-Protein Coupled Receptors. *Current Drug Targets*(8), 169-184.

Burns, L. R. (2012). *The Business of Healthcare Innovation.* Cambridge : Cambridge University Press.

George, S. R., O'Dowd, B. F., & Lee, S. P. (2002). G-Protein-coupled receptor oligomerization and its potential for drug discovery. *Nature Reviews Drug Discovery , 1*, 808-820.

GPCRDB partnership. (2007). *GPCRDB information system G protein-coupled receptors*. Retrieved January 2, 2016, from http://www.gpcr.org/7tm/

Halka, C. (2014). *Class C GPCR Metabotropic Glutamate Receptor subtype discrimination using Computational Intelligence methods.*

Hall, M., Witten, I., & Frank, E. (2011). *Data Mining: Machine Learning Tools and Techniques.* Morgan Kaufmann Publishers.

Kelller, J., Gray, M., & Givens, J. (1985). IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS . *SMC-14*(4).

Klabunde, T., & Hessler, G. (2002). Drug Design Strategies for Targeting G-Protein-Coupled Receptors. *ChemBioChem, 3*(10).

Kohavi, R. (1995). A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. International Joint Conference on Arti.

König, C., Cárdenas, M. I., Giraldo, J., Alquézar, R., & Vellido, A. (2015). Label noise in subtype discrimination of class CG protein-coupled receptors: A systematic approach to the analysis of classification errors. *BMC bioinformatics*, 16-314.

Konig, C., Vellido, A., Alquezar, R., & Geraldo, J. (2014). Finding class C GPCR subtype-discriminating n-grams through feature selection.

Krantz, B. (n.d.). Principles of Metabolic Regulation, Illustrated with Glucose and Glycogen Metabolism.

Li, N. P. (2002). The Molecule Pages database. *Nature*.

Mombaerts, P. (2004). Genes and ligands for odorant, vomeronasal and taste receptors. *Nature*.

Perco, P., Rapberger, R., Siehs, C., Lukas, A., Oberbauer, R., Mayer, G., et al. (2006). Transforming omics data into context: Bioinformatics on genomics and proteomics raw data. *ELECTROPHORESIS, 27*(13), 2659-2675.

Pollak M.R., B. E. (1994). Autosomal dominant hypocalcaemia caused by a Ca(2+)-sensing receptor gene mutation.

Ronald S Duman, G. K. (2016). Synaptic plasticity and depression: new insights from stress and rapid-acting antidepressants. *Nature*.

Schneider, J. (1997). *Cross Validation*. Retrieved January 2, 2016, from http://www.cs.cmu.edu/~schneide/tut5/node42.html

Scikit. (2010). *Linear Discriminant Analysis & Quadratic Discriminant Analysis*. Retrieved from http://scikit-learn.sourceforge.net/0.5/auto_examples/plot_lda_vs_qda.html

Shkurin, A., & Vellido, A. (2016). Random Forests for quality control in G-Protein Coupled Receptor databases. Barcelona: 4th International Work-Conference on Bioinformatics and Biomedical Engineering, IWBBIO.

Steinbach, M., Kumar, V., & Tan, P.-N. (2004). *Introduction to Data Mining.*

Tairas, R., Fitzpatrick, M., & Ledeczi, A. (2015). *Introduction To MATLAB Programming*. Retrieved from https://www.coursera.org/course/matlab

Yang Han, I. S. (2009). Allosteric communication between protomers of dopamine class A GPCR dimers modulates activation.